REMARKS

In the Office Action of January 2, 2004, the Examiner objected to claims 7 and 12 as being dependent on a rejected base claim, but otherwise indicated that these claims are allowable. Further, the Examiner rejected claims 1-6, 8-11, and 13-21 under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,065,088 to Bronson et al. ("Bronson") and rejected claims 22-25 under 35 U.S.C. § 103(a) in view of Bronson. Regarding the rejection under 35 U.S.C. § 102(b), Applicants initially note that although the Examiner did not include claim 13 in the statement of the rejection on page 2 of the Office Action, the Examiner included this claim in the substantive rejection on page 5 of the Office Action.

By this Amendment, Applicants have amended claims 2, 21, and 24 to improve form. Claims 1-25 are currently pending.

Applicants respectfully disagree with the Examiner's rejection of claims 1-6, 8-11, and 13-21 under 35 U.S.C. § 102(b). A proper rejection under 35 U.S.C. § 102 requires that a single reference teach every aspect of the claimed invention either expressly or impliedly. Any feature not directly taught must be inherently present. See M.P.E.P. § 2131. Applicants submit that Bronson does not teach every aspect of claims 1-6, 8-11, or 13-21 either expressly or impliedly.

Claim 1, for example, recites a queue that includes a first queuing area configured to enqueue and dequeue data and a second queuing area configured to receive data from the first queuing area when the first queuing area has data available to be dequeued. Claim 1 further recites bypass logic coupled to the second queuing area, the bypass logic configured to bypass the first queuing area and to forward data to the second queuing area when the second queuing area is ready to receive data and the first queuing area is empty.

Bronson, in contrast to claim 1, is directed to systems and methods for interrupt command queuing and ordering. (Bronson, Title). As illustrated in Fig. 3, Bronson includes a command queue 136 for ordering EOI (end-of-interrupt) commands and a command queue 134 for ordering INR (Interrupt return) and IRR (interrupt reissue request) commands. (Bronson, Fig. 3 and col. 8, lines 9-54). Additionally, Bronson discloses a FIFO command queue 146 that orders bus memory mapped input output (MMIO) commands. (Id.)

Applicants submit that Bronson fails to disclose or suggest a number of the features recited in claim 1. Claim 1, for example, recites first and second queuing areas and bypass logic. The bypass logic is configured to "bypass the first queuing area and to forward data to the second queuing area when the second queuing area is ready to receive data and the first queuing area is empty." Although Bronson may disclose a number of queues, Bronson does not disclose or suggest the bypass logic recited in claim 1.

The queues in Bronson instead appear to be designed to prioritize different type of commands. More specifically, queue 134 of Bronson receives INR and IRR commands. Queue 136 receives EOI commands. The EOI commands are eventually dequeued from queue 136 and added to command queue 146 with incoming MMIO commands. This system is said "to enforce strict ordering of EOI commands relative to MMIO accesses." (Emphasis added) (Bronson, col. 8, lines 13-15). The INR and IRR commands, however, are dequeued to high priority output queue 150, which is said to allow "INR and IRR commands to bypass enqueued MMIO access." Thus, to summarize the sections of Bronson cited by the Examiner, Bronson prioritizes the queueing of commands, in which some types of commands are maintained in a strict queue order and other types (i.e., INR and IRR commands) are sent to a high priority queue.

Prioritizing commands, as disclosed by Bronson, does not disclose or suggest the bypass logic of claim 1. In particular, the bypass logic of claim 1 bypasses the first queuing area when the second queuing area is ready to receive data and the first queuing area is empty. The "bypassing" disclosed by Bronson, however, appears to be performed for every INR and IRR command, regardless of the state of another queuing area. The EOI commands, in contrast (queue 136 of Bronson), are not bypassed at all. Instead, commands from this queue appear to be always forwarded to queue 146.

The Examiner contends, regarding the bypass logic recited in claim 1, that the "the queuing system also has the ability to bypass the first queuing area by sending specific commands directly to the command queue in the second queuing area through bus 141." (Office Action, paragraph 3). The Examiner appears to be stating that because the MMIO commands are sent directly to command queue 146, that these commands effectively "bypass" the queues in interrupt router 142. Applicants respectfully disagree with this interpretation of "bypass." The MMIO commands of Bronson appear to be always sent to command queue 146. The queues in interrupt router 142 do not appear to ever be used to process these commands. The bypassing recited in claim 1, in contrast, recites specific conditions under which the first queuing area is bypassed and data is forwarded to the second queuing area.

For at least these reasons, Applicants submit that Bronson does not disclose or suggest these features of claim 1, as required by 35 U.S.C. §102(b). Accordingly, the rejection of claim 1 should be withdrawn.

Independent claim 8 is directed to a method of masking latency in a queue. The method includes receiving incoming data items for the queue, forwarding the incoming data items to a buffer when the queue is empty and the buffer is free to receive data items, enqueuing the incoming data items in the queue when the queue contains data items or the buffer is not free to receive data items, dequeuing data items from the queue to the buffer when the buffer is free to receive data items, and transmitting the data items from the buffer as the output of the queue.

In rejecting claim 8, the Examiner relies on queues 146 and 148 of Bronson (Fig. 3 of Bronson), and states:

> Bronson et al. teaches a FIFO command queue 146 receiving memory mapped input/output (MMIO) commands from a system bus control logic 140. From common knowledge of how FIFO queues work (Microsoft Computer Dictionary, Page 187), it is understood that the data items in the command queue 146 are forwarded to the normal priority queue 148 ("buffer") via bus 147 when the command queue is empty and the normal priority queue has space for the data item. In addition, it is understood that in the instance when the command queue is not empty, new data items are simply added to the top of the queue as older data items are de-queued and transferred to the normal priority queue once space is allocated in the normal priority queue.

(Office Action, page 3). Applicants strongly disagree with what the Examiner simply assumes is "understood" based on the cited Microsoft Computer Dictionary. The Microsoft Computer Dictionary, at page 187, defines "first in, first out" as "A method of processing a queue, in which items are removed in the same order in which they were added—the first in is the first out." Applicants submit that nothing in this definition of "first in, first out" discloses or suggests forwarding incoming commands, such as MMIO commands on bus 141 or EOI commands on bus 139, to a buffer when the command queue 146 of Bronson is empty. If anything, this definition simply implies that incoming items are added to command queue 146, processed through command queue 146 in FIFO order, and then input to queue 148. This interpretation of Bronson is supported by the disclosure of Bronson. For example, in Fig. 3, the incoming MMIO and EOI commands are input to MUX 144 and then to command queue 146 via line 145. Similarly, at the other end of command queue 146, line 147 leads to the input of queue 148. This aspect of Bronson is further disclosed at col. 8, lines 30-32 of Bronson: "The output of command queue 146 is fed on line 147 to normal priority output queue 148 and thence on line 149 to bus control logic 152 and I/O bus 102."

For at least these reasons, Applicants submit that Bronson fails to disclose or suggest a number of features of claim 8, including, for example, receiving incoming data items for a queue, forwarding the incoming data items to a buffer when the queue is empty and the buffer is free to receive data items, and enqueuing the incoming data items in the queue when the queue contains data items or the buffer is not free to receive data items. Thus, the rejection of claim 8 under 35 U.S.C. § 102(b) is improper and should be withdrawn.

Independent claim 20 was rejected by the Examiner using rationale similar to that applied in rejecting claim 8. Accordingly, for reasons similar to the reasons given above, Applicants submit that the rejection of independent claim 20 is also improper under 35 U.S.C. § 102(b) and should be withdrawn.

Independent claim 14 was also rejected under § 102(b) by the Examiner based on Bronson. Claim 14 recites, among other things, "a request manager configured to receive memory requests and a plurality of parallel processors configured to receive the memory requests from the request manager." Bronson does not disclose the claimed request manager and the plurality of processors recited in this claim. System bus 100 of Bronson transmits commands to system bus control logic 100, which then forwards the commands to queues, such as queue 146. Bronson, however, completely fails to disclose or suggest a request manager and a plurality of parallel processors configured to receive the memory requests from the request manager, as recited in claim 14.

Claim 14 further recites a queue corresponding to each of the plurality of parallel processors, each of the queues configured to enqueue and dequeue memory requests of the corresponding parallel processor, and a buffer configured to receive memory requests dequeued from the queues when the queues contain memory requests and to receive memory requests directly from the input port when the queues do not contain memory requests. The Examiner appears to rely on command queue 146 and normal priority queue 148 of Bronson to allegedly disclose these features of the invention. (Office Action, page 3). Applicants disagree with the Examiner's interpretation of Bronson. As explained previously, Bronson discloses that incoming commands are added to command queue 146, processed through command queue 146 in FIFO order, and then input to queue 148. Nothing in Bronson, however, discloses or suggests a buffer configured to receive memory requests dequeued from the queues when the queues contain memory requests and to receive memory requests directly from the input port when the queues do not contain memory requests, as recited in claim 14.

For at least these reasons, Applicants submit that the rejection of claim 14 in view of Bronson under 35 U.S.C. § 102(b) is improper and should be withdrawn.

Claims 2-6, 9-11, 13, 15-19, and 21 variously depend, either directly or indirectly, from one of independent claims 1, 8, 14, or 20. At least by virtue of their dependency on an independent claim, Applicants submit that the rejection of these dependent claims should be withdrawn.

Claims 2-6, 9-11, 13, 15-19, and 21 recite additional features that are not disclosed or suggested by Bronson. Claim 2, for example, as amended, recites that the data consists of memory access requests. Bronson, in contrast, clearly discloses that the system shown in Fig. 3 handles a number of different types of commands, including EOI commands, IRRs, and INRs. (see Bronson, col. 8, lines 8-19). Claim 21 includes features similar to claim 2. Accordingly, Bronson does not disclose or suggest the features recited in this claim as required by 35 U.S.C. § 102(b).

Claims 22-25 stand rejected under 35 U.S.C. § 103(a) in view of Bronson. Applicants respectfully traverse this rejection.

Claim 22 is directed to an arbiter comprising a queue, a multiplexer, and arbitration logic. The queue is configured to enqueue data items at a first stage of a plurality of stages and dequeue the data items at a last stage of the plurality of stages of the queue. The multiplexer has a plurality of inputs connected to different stages of the queue. The multiplexer outputs selected ones of the data items read from the queue. The arbitration logic is coupled to the queue and controls the multiplexer to output the selected ones of the data items by selecting a predetermined number of data items from the queue during an arbitration cycle, the arbitration logic giving higher priority to data items in later stages of the queue.

The Examiner, in rejecting claim 22, contends that Bronson discloses a plurality of queues, but concedes that Bronson does not "teach the use of a multiplexer connected to multiple stages of a queue, outputting selected data items, and coupled to and controlled by the I/O bus control logic 152." (Office Action, paragraph spanning pages 5 and 6). The Examiner, however, contends that this would have been an obvious modification "since multiplexers are shown to be used in the selection of signals (Figure 3, reference #144) and such a component would be useful in selecting signals from either bus 149 or bus 151 (Figure 3)." (Office Action, page 6).

Applicants submit that the Examiner has not made a proper *prima facie* case of obviousness under § 103. Although Bronson does disclose a multiplexer 144, multiplexer 144 is not configured like the multiplexer recited in claim 22. In stark contrast, multiplexer 144 is explicitly shown as connecting only to the input of command queue 146. Bronson is completely devoid of any disclosure or suggestion to modify the multiplexer of Bronson as suggested by the Examiner. The fact that a multiplexer can be used to "select signals" in no way discloses or suggests the multiplexer of claim 22, which includes a multiplexer that "has a plurality of inputs connected to different stages of the queue." Applicants submit that the Examiner, in making the rejection of claim 22, is impermissibly using hindsight gleaned from Applicant's specification.

For at least these reasons, Applicants submit that the rejection of claim 22 under § 103 is improper and should be withdrawn. The rejection of claims 23-25, at least by virtue of their dependency on claim 22, is therefore improper and should also be withdrawn.

Claims 23-25 recite additional features that are not disclosed or suggested by Bronson. Claim 23, for example, further defines the arbiter of claim 22, and recites that the arbiter further comprises bypass logic coupled to the queue, the bypass logic causing the data items to bypass the queue and to forward the data items to the multiplexer when the queue is empty. Based on rationale similar to that given above regarding claim 1, Applicants submit that Bronson also does not disclose or suggest this feature of the invention as required by § 103.

Additionally, claim 24, as amended, recites that the data items consists of memory access requests. Bronson, in contrast, discloses that the system shown in Fig. 3 handles a number of different types of commands, including EOI commands, IRRs, and INRs. (see Bronson, col. 8, lines 8-19).

In view of the foregoing amendments and remarks, Applicants respectfully request the Examiner's reconsideration of this application, and the timely allowance of the pending claims.

To the extent necessary, a petition for an extension of time under 37 CFR 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 50-1070 and please credit any excess fees to such deposit account.

Respectfully submitted,

HARRITY & SNYDER, L.L.P.

By: ______________________
Brian E. Ledell
Reg. No. 42,784

11240 Waples Mill Road
Suite 300
Fairfax, Virginia 22030
(571) 432-0800
Customer Number: 26615

Date: 4/2/04